Coupling Friction with Visual Appearance

SHELDON ANDREWS, École de technologie supérieure, Canada
LOIC NASSIF, McGill University, Canada
KENNY ERLEBEN, University of Copenhagen, Denmark
PAUL G. KRY, McGill University, Canada

We present a novel meso-scale model for computing anisotropic and asymmetric friction for contacts in rigid body simulations that is based on surface facet orientations. The main idea behind our approach is to compute a direction dependent friction coefficient that is determined by an object's roughness. Specifically, where the friction is dependent on asperity interlocking, but at a scale where surface roughness is also a visual characteristic of the surface. A GPU rendering pipeline is employed to rasterize surfaces using a shallow depth orthographic projection at each contact point in order to sample facet normal information from both surfaces, which we then combine to produce direction dependent friction coefficients that can be directly used in typical LCP contact solvers, such as the projected Gauss-Seidel method. We demonstrate our approach with a variety of rough textures, where the roughness is both visible in the rendering and in the motion produced by the physical simulation.

CCS Concepts: • **Computing methodologies → Physical simulation**.

Additional Key Words and Phrases: physical simulation, contact, friction, normal mapping, GPU techniques

1 INTRODUCTION

Comprehensive models of friction are relevant for contemporary interactive computer graphics applications. For instance, friction is an operational mechanism in virtual reality, games, and training simulations, and is important for scenarios such as stacking and wheel-terrain interaction. Yet models that capture a wide range of friction phenomena remain elusive, at least in the context of real-time simulations.

The process of tuning friction to achieve specific effects can be tedious as simulation designers are often limited to adjusting values in a table of pair-wise material constants for isotropic Coulomb friction. Consequently, recent work in computer graphics has explored new approaches to reduce the combinatorial nightmare of tuning this parameter table of all possible material and surface-finish combinations. Specifically, friction tensors [Pabst et al. 2009] and the matchstick friction model [Erleben et al. 2020] both use a parameterization that combines surface information to derive the frictional properties dynamically at any contact. Our work is similar in that we also compute custom frictional properties based on the surfaces' configuration at each contact. However, a unique

Authors' addresses: Sheldon Andrews, École de technologie supérieure, Montreal, Canada, sheldon.andrews@etsmtl.ca; Loic Nassif, McGill University, Montreal, Canada, loic.nassif@gmail.com; Kenny Erleben, University of Copenhagen, Copenhagen, Denmark, kenny@di.ku.dk; Paul G. Kry, McGill University, Montreal, Canada.
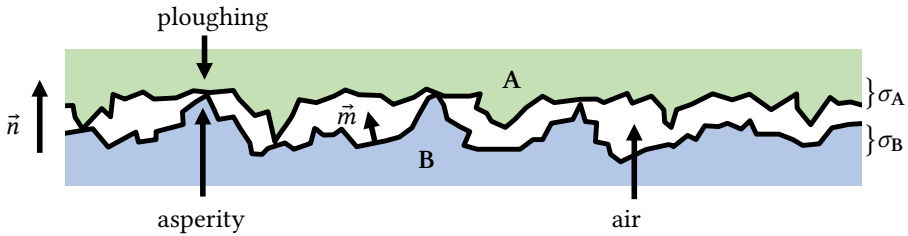
Fig. 1. Illustration of contact between surfaces **A** and **B**, with macroscopic (i.e., mesh geometry) normal $\vec{n}$ and meso-facet normals $\vec{m}$, where facet variance σ can be used to denote overall roughness. We model friction in the regime of interlocking, where ploughing models and meso-facet orientation can be used to explain the coefficient of friction.

aspect of our approach is that we use small scale surface geometry to compute a plausible friction response. Effectively, we couple the visual appearance of surface roughness to the friction model.

True friction is a complicated system response combining many phenomena at different scales [Vakis et al. 2018]. Ignoring thermal effects, chemical reactions, lubrication, and wear, the friction between hard rigid surfaces is largely considered to be primarily related to their roughness. There are several ways to measure and encode surface roughness. For instance, amplitude can be used to characterize vertical deviations from the mean surface, whereas other measures of roughness are based on the variation of the surface slope. In this work, our definition of roughness follows directly from the latter in that we encode roughness with varying surface normals.

The computer graphics community has developed a variety of methods for rendering rough surfaces, including bump maps, normal maps, BRDFs, and other micro-facet models. In this paper, we are motivated by the challenge of bringing rendered surface roughness and friction effects into agreement while keeping the performance of the approach suitable for interactive applications, such as VR or video games. Specifically, we compute friction coefficients from normal maps that provide the meso-level roughness detail.

Figure 1 shows a diagram of rough surfaces in contact, where small visible salient features (asperities) on one surface come into contact with the other, and the interaction of these is a principal factor that explains the variation of available friction forces. In contrast to classic tribology models that estimate friction from interlocking micro-geometry, our work includes a directional component which leads to anisotropic and asymmetric friction cones that are useful in the simulation frameworks commonly used in computer graphics. Our work addresses *meso*-level phenomenological modeling of salient and visible geometric features. That is, at a scale where geometric features due to surface roughness become visibly distinct.

We highlight the main properties of our model below:

Proportionality. Friction forces increase with surface roughness. Although this is a broad observation, it is generally true that rougher surfaces will generate more friction.

Anisotropy. When small scale geometry variation on one of the surfaces exhibits anisotropy, meaning that normal deviations are larger in one direction compared to another, then our friction model will also produce anisotropic behavior. That is, the friction forces will depend on the orientation of the surfaces at the contact, as well as the relative sliding direction.

Asymmetry. Surface geometry can interlock when moved in one direction, but begin to separate for motion in the opposite direction. In such examples, our model generates larger friction forces in the former case, and smaller friction forces in the latter case.

Visual consistency. The primary benefit of our approach is the ability to model and set friction directly from the same representation used for visual rendering of the surface. This can be done procedurally, such as from statistical distributions of facets as is done in global illumination, or by encoding the facet geometry explicitly, such as by a displacement field or normal map.

In this work, we compute friction forces by generating a limit surface approximation (i.e., a friction cone or box) that is obtained by extracting local information at each contact using a novel process that leverages the GPU. The limit surface can then be used with simulators based on both direct and iterative linear complementarity (LCP) solvers and is independent of the time-stepping scheme, following the methods of Erleben et al. [2020]. Furthermore, we specifically demonstrate that our friction model may be used in conjunction with the prolific normal mapping technique [Cohen et al. 1998], and show that a range of compelling simulations may be realized using explicit surface normal information. The resulting frictional behavior matches qualitatively the visual rendering and exhibits the roughness dependence we target in our approach.

2 RELATED WORK

The recent course notes by Andrews and Erleben [2021] provides a good overview of simulation methods and numerical methods for solving friction problems. A comparison of current friction models in computer graphics is also presented by Erleben et al. [2020]. Below, we discuss other related work in graphics as well as those in mechanical engineering and tribology.

2.1 Friction in computer graphics

Asymmetric and anisotropic friction is found in many places in our everyday surroundings, and is unlike the isotropic Coulomb friction model that is employed in the majority of computer graphics simulations [Allard et al. 2010; Baraff 1994; Daviet 2020; Daviet et al. 2011; Erleben 2017; Erleben et al. 2020; Geilinger et al. 2020; Kaufman et al. 2005, 2008; Li et al. 2020; Ly et al. 2020; Macklin et al. 2019; Otaduy et al. 2009; Peiret et al. 2019]. Nevertheless, there is a collection of work in computer graphics dedicated to anisotropic and nonlinear models such as friction tensors [Pabst et al. 2009], cloth friction [Chen et al. 2013], and the matchstick model [Erleben et al. 2020]. In contrast, our approach starts at a lower level, using principles of roughness and asperities inspired from real-world surfaces. Furthermore, we make a strong connection between the visual appearance of surfaces and their simulated motion.

The roughness of real world surfaces is also a critical component of acoustic and haptic rendering. Otaduy et al. [2004] combine height field texture information on two contacting objects to give a perception of roughness, while Costes et al. [2018] encode additional haptic features into texture maps for use in real-time haptic rendering. For contact sounds, van den Doel et al. [2001] use a one dimensional roughness computed from a fractal noise model or measured by microphone contact. Ren et al. [2010] extend this idea to also use two dimensional texture, which are visible to the user but not to the physics engine, as part of their contact sound generation. In contrast, making roughness information visible to the physics engine is our primary goal.

2.2 Mechanical engineering and robotics

The mechanical engineering community has investigated more general models for friction. Liley et al. [1998] investigate anisotropies and asymmetries with lateral force microscopy in the wearless regime. Yu and Wang [2012] examine the relationship between anisotropic friction and surface roughness. Umbanhowar et al. [2012] observe non-convex friction responses on vibrating plates, and Walker and Leine [2017] propose ways to deal with non-convex friction models. This line of work demonstrates an increasing interest in the engineering community for more expressive

friction models. However, this work is mainly concerned with measurement for specific scenarios at the micro-scale, which is lower than what is visible in many computer graphics applications. In contrast, our focus is on general models that are easy to rig in the context of interactive graphics applications and provide consistency with the visual appearance of rough surfaces.

Other work has explored measurement-based modeling of surface roughness using robotic apparatus. Pai et al. [2001] record contact friction textures using a robotic probe under the assumption of a symmetric isotropic Coulomb model. Dreßel et al. [2019] perform model free friction measurements with a robot arm with limited ability to sample randomly, and focus on measuring force rather than explaining the cause.

2.3   Tribology

In the field of tribology [Rabinowicz 1995; Sheng Chen and Liu 2016], friction at the microscopic scale has been studied for a long time and many models exist in this regime. A key concept applied by many models is that the friction force is related to the real area of contact between two surfaces. The notion of *real* versus *apparent* contact area can be explained using Figure 1. Consider that two rough surfaces are like mountain ranges, with one of them being turned upside-down and pushing against each other. The real contact area is due to the regions where the peaks of mountains (asperities) touch the other surface, whereas the apparent area is due to the land (macro) surface.

In seminal work by Bowden and Tabor [1950], they observed that the real area of contact is proportional to the normal load and independent of the apparent area of contact. This has given rise to models that consider deformation of asperities forming an adhesive binding that contributes to the friction. Greenwood and Williamson [1966] use probabilities for asperity contact to derive their model. When considering continuous sliding motion, asperities start to plow through the surfaces and so comes a larger contribution to the friction effect. This is the domain of application we seek for computer graphics applications: very rough surfaces with salient visible features at the meso-scale that introduce changes and variation to an object's motion. Our model borrows inspiration from tribology in that we want the coefficient of friction to be a monotonically increasing function of roughness measure larger than 2 microns. This is the region where friction is affected by asperity interlocking [Rabinowicz 1995] and where the model of conical asperity ploughing has been proposed for computing a coefficient of friction depending on the slope of the conical asperity [Sheng Chen and Liu 2016].

Tribology models are not always directly applicable to our domain as we seek meso-scale models that can account for asymmetry and anisotropy. Furthermore, we seek a model that falls naturally into the machinery of complementarity based modeling of contact [Andrews and Erleben 2021; Bender et al. 2014]. Hence, we glue concepts together to create a novel approach for generating friction cones from these first principles.

3   FRICTION MODEL

There are numerous factors that determine the friction between dry solids, such as the real area of asperity contact, the shear and fracture strength of the material, and other elements such as temperature and adhesion.

Our model focuses on friction effects that arise between rough surfaces due to the slope of small surface facets. Specifically, our goal is to capture effects due to interaction at the meso-scale, where rough features start to become visible to the human eye. At this level, friction is mainly due to asperity deformations and ploughing. As seen in Figure 2, the meso-scale regime corresponds to a roughness RMS measure larger than 2 microns. For a region of lower roughness values, friction is constant, and eventually for very smooth surfaces the friction increases due to larger contact areas and molecular bonding between deformed asperities. However, we only focus on the regime where
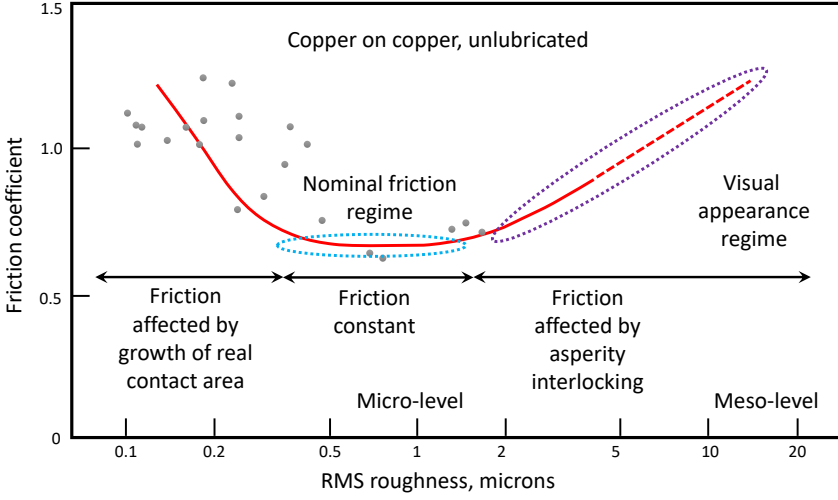
Fig. 2. The effective friction coefficient due to surface roughness. Adapted from Rabinowicz [1995], the plot is extrapolated (dashed red) into the visual appearance regime targeted by our work.

larger and possibly visible features influence the friction behavior. In this regime, the plot suggests the use of a nominal constant friction coefficient, which grows as the RMS roughness increases above 2 microns due to interlocking meso-scale features.
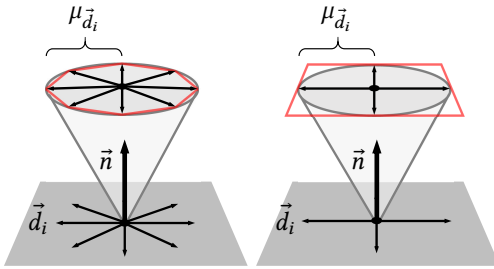

Fig. 3. Sampling in directions $\vec{d}_i$ to generate a polyhedral approximation (left) and box approximation (right) of the friction cone.

Since the real contact area between rough surfaces makes up only a tiny fraction of the apparent contact area [Rubinstein et al. 2006], we assume it to be unlikely that asperities from both surfaces occur at the same location. This suggests that friction due to a pair of contacting surfaces has an additive nature, at least for rigid surfaces with a meso scale roughness profile and ignoring plastic deformation and fracturing of asperities. Therefore, we assume that the coefficient of friction in the direction $\vec{d}$ can be computed by an additive model, such that

$$\mu_{\vec{d}} \approx \mu_{\vec{d},\mathbf{A}} + \mu_{\vec{d},\mathbf{B}} \,. \tag{1}$$

Here, $\mu_{\vec{d},\mathbf{A}}$ and $\mu_{\vec{d},\mathbf{B}}$ encode the scaling with respect to surface roughness and direction effects stemming from salient meso-scale features from each surface. We use a modulated multiplicative form to model this effect involving two terms, such that

$$\mu_{\vec{d},\mathbf{A}} \approx \mathcal{S}_{\vec{d},\mathbf{A}} \cdot \mathcal{P}_{\vec{d},\mathbf{A}} \,, \tag{2}$$

where $\mathcal{S}$ is a function that scales friction depending on how much contact is made between surfaces, and $\mathcal{P}$ models the directional effect depending on the surface meso-geometry. Hence, the coefficient $\mu_{\vec{d},\mathbf{A}}$ depends on the meso-facet normal $\vec{m}$ and the macroscopic surface normal $\vec{n}$ of surface $\mathbf{A}$, and a similar form applies to $\mu_{\vec{d},\mathbf{B}}$.

We target sampling strategies suitable for interactive simulation and one can apply various sampling strategies for $\vec{d}$ to generate friction cone approximations that span the contact plane. Figure 3 illustrates this concept. Experiments in this work use the box cone approximation. However,

polyhedral cone approximations are likewise compatible with our proposed model. Similarly, the specific form of the $\mathcal{S}$ and $\mathcal{P}$ functions and how they are computed over a surface patch is tailored to exploit computer graphics techniques for geometric modeling and rendering.

We demonstrate how roughness represented by a normal map may be used to efficiently compute coefficients that are suitable for use in friction simulation by averaging over small regions of the surface using a process we later explain in Section 4. In this section, we provide details about our friction model. For increased readability in our subsequent derivation we will omit subscripts $\vec{d}$ and surfaces as we consider a single direction and surface only. Later we add subscripts back when assembling all terms to present our full model.

3.1 Ploughing model of friction

Micro-scale ploughing occurs when asperities on one surface dig into or grind against the material on another surface. This phenomena is demonstrated in Figure 4. Assuming a conical asperity, the friction due to ploughing is a function of the effective slope of the asperity. Computing the angle of the slope, let

Fig. 4. Ploughing friction.

$$\theta = \mathrm{acos}(\vec{n} \cdot \vec{m}),$$

where $\vec{n}$ is the macro normal of the underlying geometry (e.g. from a polygon mesh), and $\vec{m}$ is the micro-facet normal due to the asperity. The friction coefficient due to a surface ploughing against another is then computed as

$$\mu = \frac{2}{\pi}\tan\theta. \tag{3}$$

The above equation is a commonly accepted ploughing model of friction that can be found in standard references on contact dynamics [Sheng Chen and Liu 2016]. However, it is a 1D model based on a cross-sectional analysis of the surfaces.

Consider the illustration in Figure 4. It is clear that some friction is produced as the green surface moves in the direction $\vec{d}$ and interlocks with the blue surface since $\vec{m}$ is oriented in the ploughing direction. Observe also that the $\tan\theta$ term increases rapidly the more that the micro-facet normal is directly aligned with the ploughing motion. However, for an out-of-plane motion there may be no resistance since the micro-facet normal is not oriented in the direction of ploughing. Yet we seek a friction model that is valid for cases where the facet normal and ploughing direction do not lie in the same plane. Therefore, we use Equation 3 as a building block and extend it to a directional meso-scale version that accounts for the friction resistance produced for any motion direction, $\vec{d}$, in the contact plane.

3.2 Directional friction

In 3D, we can intuitively think of the many micro-scale asperities as working like a "meso-scale" blade of a snow plough, where the meso-facet orientation now includes both the slope of the plough and the ploughing direction as illustrated in Figure 5. Thus, the coefficient computed by Equation 3 should apply in the maximal direction, and should drop to zero in the perpendicular direction. This relationship is well defined by the cosine of the angle between $\vec{d}$ and the meso-facet normal $\vec{m}$ in the contact plane.

The coefficient computed in Equation 3 can then be modulated according to the orientation of a meso-facet relative to the maximum friction direction, giving the modified version of the coefficient
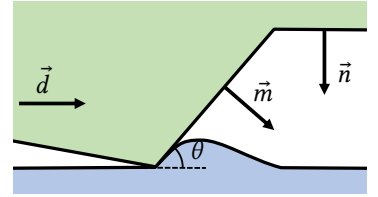
computation:

$$\mu = (\vec{m}' \cdot \vec{d}) \frac{2}{\pi} \tan\theta, \quad \text{where } \vec{m}' = \frac{\vec{m} - (\vec{m} \cdot \vec{n})\,\vec{n}}{\|\vec{m} - (\vec{m} \cdot \vec{n})\,\vec{n}\|} \,. \tag{4}$$

Here, $\vec{m}'$ is $\vec{m}$ projected onto the contact plane, which lets all attenuation due to meso-facet slope be handled by the $\tan\theta$ term.

3.3 Asymmetric friction

Equation 4 encodes the anisotropic nature of friction due to varying facet orientations. However, it fails to capture that a meso surface may be oriented away from the movement direction. For instance, consider the scenario depicted in Figure 4. If the movement direction were reversed, in this case we expect that the meso facet will contribute less to the friction effects. This example highlights the asymmetric nature of friction due to interlocking meso features. Based on this observation, we further extend the friction model to encode the one-sided nature of the behavior using the max function:

Fig. 5. For directional friction in 3D, we assume the meso-facet exists as a narrow wedge, and maximal friction occurs in the direction $\vec{d}_1$, i.e., the direction of the meso-facet $\vec{m}$ projected onto the plane with normal $\vec{n}$. No resistance occurs in the orthogonal direction $\vec{d}_2$.

$$\mu = \underbrace{\max(0, \vec{m}' \cdot \vec{d})}_{\mathcal{S}} \underbrace{\frac{2}{\pi} \tan\theta}_{\mathcal{P}} \,. \tag{5}$$

Observe that we have now recovered the physical model we sought in Equation 2. Since we intend to use our friction model in the context of interactive simulations, we compute the friction coefficient between two colliding bodies by using the sum of coefficients computed using facets on each surface. At a specific contact point, the coefficient is

$$\mu = \underbrace{\max(0, -\vec{m}'_{\mathbf{A}} \cdot \vec{d}) \frac{2}{\pi} \tan\theta_{\mathbf{A}}}_{\mu_{\vec{d},\mathbf{A}}} + \underbrace{\max(0, \vec{m}'_{\mathbf{B}} \cdot \vec{d}) \frac{2}{\pi} \tan\theta_{\mathbf{B}}}_{\mu_{\vec{d},\mathbf{B}}} \,, \tag{6}$$

where $\theta_{\mathbf{A}} = \text{acos}(-\vec{n} \cdot \vec{m}_{\mathbf{A}})$ and $\theta_{\mathbf{B}} = \text{acos}(\vec{n} \cdot \vec{m}_{\mathbf{B}})$. Note that the sign change for the dot product terms involving the first body is because we use the same macro normal $\vec{n}$ for the two surfaces, which is generated by collision detection. This is the same contact normal used to construct non-interpenetration constraints in our simulations, and thus the sign change is necessary since we use the convention that the contact normal always points from the second body towards the first body.

3.4 Encoding of salient meso features

Our salient meso scale features originate from the idea of visual ridges, cracks and spikes that stick out of the overall flat surface and cause interlocking, asymmetry and anisotropic effects. To have a model that fits into most common graphics rendering pipelines, we use normal maps to partly encode this meso level geometry into our model. However, normal maps are limited in that they only tell us the meso facet orientation and carry no information about facet surface height. Alternatively, to represent the meso geometry of a surface, one may encode surface roughness in terms of displacement maps. However, we take the normal map approach to maintain compatibility with typical rendering pipelines for interactive graphics applications.
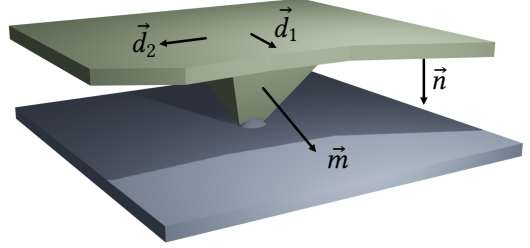
We introduce field masks, k_A and k_B, to accentuate salient meso features on the surface. They are assigned independently and provide the added benefit of permitting artistic adjustments. In our examples, k_A and k_B are simply constants that are specified per object, but they could also be stored alongside the normal texture encoding, e.g. as coefficients derived from edge filter detection of the texture image.

Furthermore, we supplement our model with a nominal friction coefficient μ_0 to model the constant friction behavior observed in the plot of Figure 2. Note that this is an ambient global setting and not a material pair-wise constant. Thus, the final version of our friction model is:

$$\mu_{\vec{d}} = \mu_0 + k_A\, \mu_{\vec{d},A} + k_B\, \mu_{\vec{d},B} \,. \tag{7}$$

Next, we provide details on how to compute an aggregate friction coefficient by evaluating Equation 7 over the contact surface.

4 SIMULATION PIPELINE

Until now, we have considered friction due to a meso-facet at a specific point on a surface. However, friction due to roughness is the result of many facets interacting over a surface region. In this section, we explain how to efficiently compute friction based on surface normal variations surrounding each contact point. We choose this *contact patch* approach over alternatives such as integrating over the whole contact surface to favor easy integration into existing interactive rigid body simulators. Clearly, a contact area can be sub-sampled with interior contact points and our patch approach can hence be seen as a "disk" sampling of the contact area. In reality, most rigid body simulators only return the corner points of the contact areas due to performance and homogeneous friction settings. Our results show that with a sufficient patch extent one can still capture the desired asymmetric and anisotropic behaviors.

Texture and friction information at each patch is computed by leveraging the GPU rendering pipeline and projecting surface geometry onto a contact centric viewing plane and rendering the geometry, along with the mapped normal texture, into small images stored in an offscreen frame buffer. We then process pairs of these images to compute an aggregate friction coefficient for each contact patch. Furthermore, we demonstrate that the resulting coefficients may be used as part of a fixed-point iterative solver, a popular method used by many interactive physics simulations for solving contact LCPs.

4.1 Rendering contact textures

We begin by constructing an orthographic viewing volume around each contact point with the goal of projecting the surface geometry surrounding the contact onto the viewing plane at the center of the volume. The constructed viewing volume is illustrated in Figure 6.

The volume is oriented so that the forward viewing direction is aligned with the contact normal, $\vec{n}$, and the top-bottom and left-right axes are aligned with the tangent and binormal basis vectors $\vec{t}$ and $\vec{b}$, respectively, that span the contact plane. This results in a viewing plane that is identical to the contact plane.

Since only a small contact patch is used, the viewing volume is similarly small. This allows much of the surface
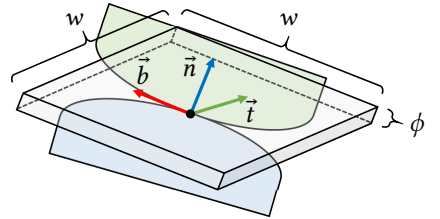

Fig. 6. An orthographic viewing volume constructed for a patch around a contact point between two surfaces. The forward direction of the viewing volume is aligned with the contact normal $\vec{n}$, and the near and far clipping plane are set according to the penetration depth ϕ of the contact.
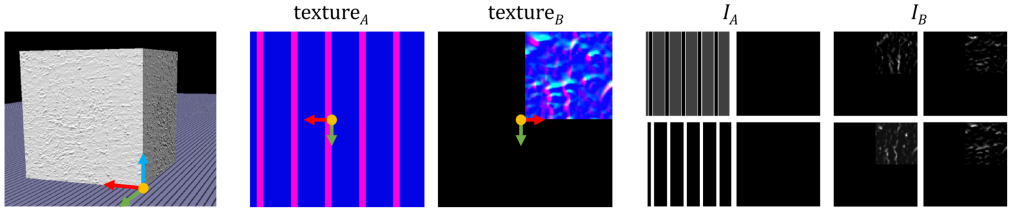
geometry to be clipped. The constructed viewing volumes have a fixed size, w, for the width and height, which is a manually adjusted global parameter for each simulation. However, the depth

Fig. 7. Left: A contact between a box with rough texture and a plane with ridges. Middle: texture$_A$ and texture$_B$ show the rendered normals from contact patches of the plane and the box. Right: Visualization of the friction coefficient sub-textures I_A and I_B generated by evaluating our friction model in four directions at each fragment.

of each specific volume is adjusted to match the penetration depth, ϕ, of the contact point. This results in volume dimensions of $w \times w \times \phi$. While $\phi = 0$ is possible, it is never observed in practice. Nevertheless, to avoid issues related to small values of ϕ, we add a value equal to 1% of the width, such that our implementation effectively uses $\phi + 0.01w$ to construct the viewing volume. When contact surfaces have high curvature at the contact point, the thin viewing volume will effectively cull all but the the surface immediately around the contact.

Then, geometry from each of the contact surfaces is rendered using a fragment shader that reads the normal texture mapped to the geometry surrounding a contact point. Each surface, s_A and s_B, is rendered into a frame buffer such that a *sub-texture* of the frame buffer contains the rendered normal texture from the contact patch of each surface. The resolution of the sub-texture is low since the region of interest (i.e., the patch size) is usually much smaller than the entire surface of the object.

Figure 7 shows an example of a box in contact with a plane along with the rendered normal textures from each contact patch. Note that in the case of the box, only the upper-right corner of the sub-texture is modified since the other regions are not covered by geometry after being clipped by the viewing volume and projected onto the viewing plane. Hence, these regions will be ignored when computing friction.

4.2 Computing directional friction coefficients

Rather than simply drawing the normal textures, we use the fragment shader to evaluate our friction model for each rendered fragment in the contact patch of each surface. This allows for more efficient use of the normal textures, which are stored in GPU memory alongside the render geometry. Also, since our model is additive, it allows the contact patch of each surface to be rendered and coefficients computed independently, which further increases the efficiency of the approach.

Assuming the box model of friction illustrated on the right of Figure 3, for each patch and for each fragment, four different friction directions are sampled based on the tangent and binormal basis vectors from Figure 6, such that

$$\vec{d}_1 = \vec{t}, \quad \vec{d}_2 = \vec{b}, \quad \vec{d}_3 = -\vec{t}, \quad \vec{d}_4 = -\vec{b}.$$

The directions are sent to the fragment shader as uniform variables, along with the macro normal which is transformed using the model-view matrix of the viewing volume. Equation 5 is then evaluated for each fragment, resulting in four friction coefficients per fragment: $\mu_{\vec{d}_1}, \mu_{\vec{d}_2}, \mu_{\vec{d}_3}, \mu_{\vec{d}_4}$.

A practical aspect of our work is that these coefficients can conveniently be packed into the 32-bit integer pixel color returned by the fragment shader. We store each directional coefficient in the red, green, blue, and alpha 8-bit color channels. The result is a pair of friction coefficient sub-textures, I_A and I_B, for the contact patch of each surface. Examples of these friction sub-textures are shown on

the right in Figure 7, where the coefficients stored in each color channel are visualized as greyscale images, and the intensity of the pixel corresponds to the magnitude of the directional coefficient.

Once rendering of all contact patches is complete, the frame buffer is transferred to CPU memory. In a final step, pairs of sub-textures are processed to compute an aggregate friction coefficient for each contact.

4.3 Processing contact textures

The final friction coefficient for each direction is computed by traversing a pair of sub-textures, pixel by pixel, obtained from a contact patch. However, not all pixels of a sub-texture are covered by geometry from both surfaces. For instance, in Figure 7 we are only interested in processing pixels in the upper-right corner of each sub-texture since this is the only region covered by geometry from both surfaces. Therefore, an additional test is used to ignore any pixel locations where only one or fewer of the sub-textures has been modified by the fragment shader.

An aggregate friction coefficient, $\bar{\mu}$, is then computed for each direction by averaging the coefficients stored in the sub-textures at valid pixel locations, P, such that

$$\bar{\mu}_{\vec{d}_k} = \frac{1}{|P|} \sum_{(i,j) \in P} I_A(i,j,k) + \frac{1}{|P|} \sum_{(i,j) \in P} I_B(i,j,k), \tag{8}$$

where $I(i, j, k)$ gives the coefficient stored at pixel location (i, j) for ploughing direction k. In our implementation with four directions, indices $k = 1, 2, 3, 4$ correspond to the the red, green, blue, and alpha channels respectively.

The aggregate friction coefficients are then ready for use by the contact solver, which uses them to determine frictional force limits at each contact point.

4.4 Frictional force limits

Our simulations follow a constraint-based contact formulation in a multibody dynamics setting with the method of Lagrange multipliers being used to enforce bilateral and unilateral constraints at each time step. We direct the reader to the course notes by Andrews and Erleben [2021] for a full treatment of this topic since we only touch on details that are relevant to our technique here.

A projected Gauss-Seidel (PGS) solver is used to compute constraint forces and frictional forces. The PGS method is a popular choice for many interactive computer graphics due to its efficiency and ease of implementation. Furthermore, constraint force limits may be updated during intermediate solver steps as solution estimates for constraint variables are refined from one iteration to the next.

Specifically, we update the bounds of friction forces as a function of the friction coefficient and the normal loading. In our simulations, the normal loading is determined by the magnitude of the non-interpenetration constraint force, $\lambda_{\vec{n}}$, which applies a force to pairs of colliding bodies in order to keep them separate at the contact point, and so they may only "push" meaning that only positive values are permitted, i.e., $\lambda_{\vec{n}} \geq 0$.

Bounds on the friction forces, $\lambda_{\vec{t}_1}$ and $\lambda_{\vec{t}_2}$, are enforced by projecting their magnitude to the range determined by the directional friction coefficients $\{\mu_{\vec{d}_1}, \mu_{\vec{d}_2}, \mu_{\vec{d}_3}, \mu_{\vec{d}_4}\}$ and the corresponding non-interpenetration force $\lambda_{\vec{n}}$, such that

$$\lambda_{\vec{t}} \leftarrow \max\left(-\bar{\mu}_{\vec{d}_3} \lambda_{\vec{n}}, \ \min\left(-\bar{\mu}_{\vec{d}_1} \lambda_{\vec{n}}, \lambda_{\vec{t}}\right)\right), \tag{9a}$$

$$\lambda_{\vec{b}} \leftarrow \max\left(-\bar{\mu}_{\vec{d}_4} \lambda_{\vec{n}}, \ \min\left(-\bar{\mu}_{\vec{d}_2} \lambda_{\vec{n}}, \lambda_{\vec{b}}\right)\right). \tag{9b}$$

Note that Equations 9a and 9b are a realization of the so-called *box* friction model, which approximates the Coulomb friction cone by a scaled box.
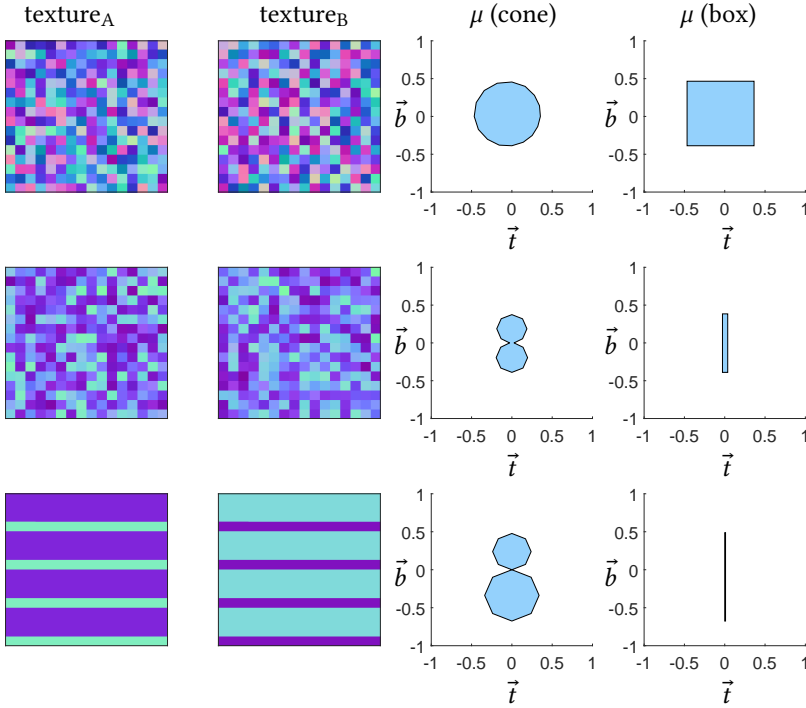
Fig. 8. Normal map textures from two surfaces (columns one and two) with their corresponding friction cone computed by our model when sampling in 16 ploughing directions (middle third column), as well as the corresponding friction box limit surface (fourth column) used in our experiments. Top row: normals generated from an isotropic Gaussian distribution with $\sigma = 1$ result in an isotropic friction cone. Middle: An anisotropic Gaussian distribution with $\sigma_x = 0.1$ and $\sigma_y = 1.0$ produces friction that is higher in vertical directions. Bottom: A sawtooth normal texture demonstrates asymmetry and anisotropy.

4.5 Synthetic contact patches

We demonstrate the friction limit surface obtained by our pipeline using the synthetically generated contact textures show in Figure 8. Note that for this test, the nominal friction coefficient has a value of $\mu_0 = 0$.

The friction coefficients of the cone limit surface are computed by sampling 16 different ploughing directions that are uniformly distributed over the unit circle, whereas the box limit surface is computed by sampling in axis aligned directions $\{-\vec{t}, -\vec{b}, \vec{t}, \vec{b}\}$. In the top row, two normal textures generated from an isotropic Gaussian distribution with $\sigma_x = 1$ and $\sigma_y = 1$ result in isotropic frictional behavior. In the middle row, the distribution is changed to an anisotropic one, with $\sigma_x = 0.1$ and $\sigma_y = 1$. Observe that there is much less variance in the red channel of the texture that corresponds to the x coordinate. As expected, this produces an anisotropic limit surface when using our model, with a vertical ploughing direction producing much higher friction compared to the horizontal direction. Finally, in the bottom row, a normal texture corresponding to a sawtooth wave is constructed and both anisotropic and asymmetric friction is observed. Moving the surfaces horizontally results in no friction, since the movement is perpendicular to the meso-facet orientation. However, movement against the steep edge of the sawtooth results in very high friction, whereas movement in the opposite direction gives slightly attenuated friction.
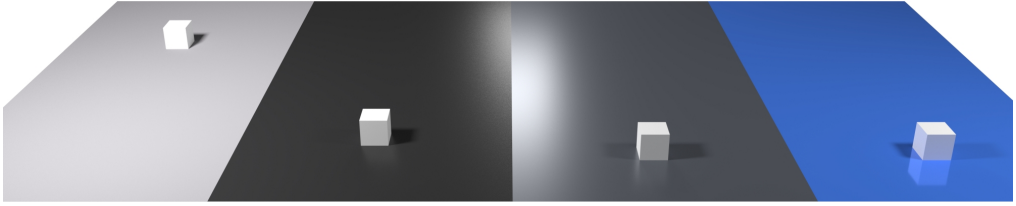
Fig. 9. Four identical boxes slide across four planes with different surface roughness. We compute friction forces based on meso-geometry details in the normal maps, and as a result the final position of the boxes depends on the surface roughness.
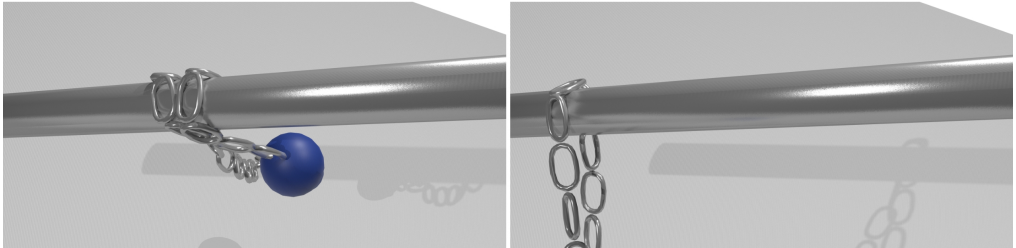

Fig. 10. A smooth rod receives a chain bolas and slips off the rod due to low friction generated from the smooth texture.
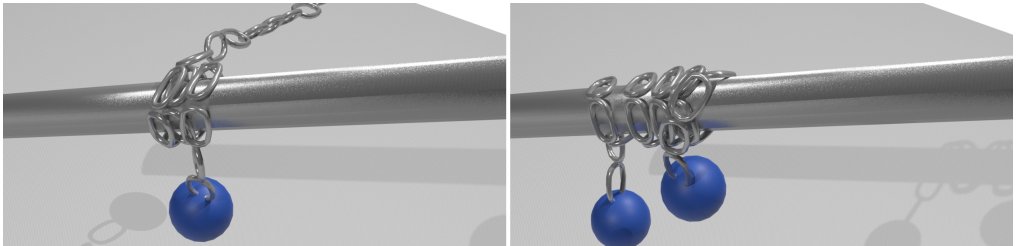

Fig. 11. A rough rod receives a chain bolas that wraps around it due to higher friction coefficients from the rough texture.

5 RESULTS

We now present examples that showcase the compelling behaviors that can be achieved by using our friction model and simulation pipeline. All experiments were performed on a PC with Intel Core i9 2.4 GHz CPU and NVIDIA GTX 1650 GPU. We mapped normal and color textures onto the surface geometry of our objects using a standard mesh editing software (Blender). A custom rigid body simulation engine was developed in C++ and used to simulate the multibody dynamics. This gave us maximum flexibility over both the simulation and the OpenGL rendering pipeline. A matrix-free version of the PGS method was implemented using the Eigen library and used to solve for constraint forces. All source code for our simulation framework, including GLSL shaders, is available in the online GitHub repository: https://github.com/sheldona/couplingFrictionVisualAppearance

Animations of the simulation experiments presented in this section can be found in the supplementary video.
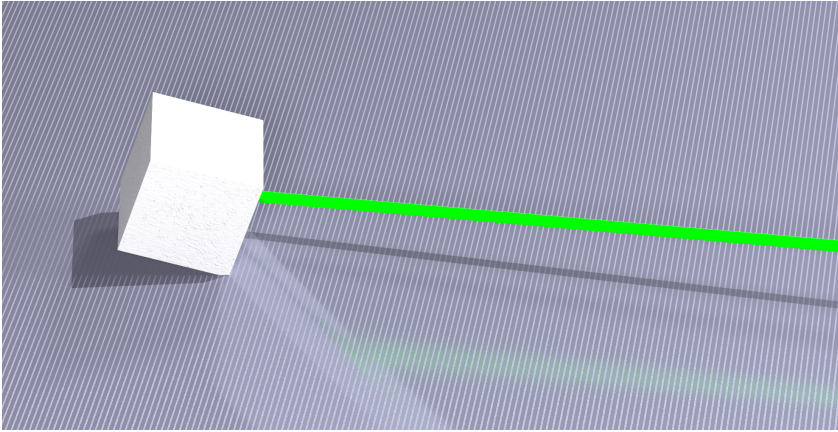
Fig. 12. A sawtooth surface texture leads to highly anisotropic and asymmetric friction. Here, a box "sticks" as it is being interactively dragged against the ridges in the texture.

5.1 Increasing roughness

Boxes on different materials. Figure 9 shows a scenario which demonstrates the ability of our model to generate increased friction with increasing roughness. In this example, the normal textures of each planar surface are created using a procedural technique where vertical height deviations (R_a) at each pixel are sampled from an isotropic Gaussian distribution function. Then, surface normal perturbations are computed at each pixel by assuming a 1×1 mm surface area for each pixel. We select average height deviations that correspond to various materials, and from left-to-right in the figure we have: concrete (R_a = 1 mm), cast iron (R_a = 0.15 mm), steel (R_a = 0.01 mm) and smooth plastic (R_a = 0 mm). All four boxes have identical mass, geometry and initial velocity, and they are textured using a normal map that resembles a smooth plaster material. As expected, the box slides furthest on the smooth plastic surface, and similar behavior is observed for the steel surface. The box on the cast iron surface comes to rest earlier, whereas the box in the concrete surface stops almost immediately after the initial impact due to high friction forces generated by the rough surface.

Chain bolas. In Figures 10 and 11, a chain bolas is thrown at a metal rod with textures of different roughness. We use some of the same textures from the *previous* example. In the case of the smooth rod (R_a = 0.01 mm), the chain is unable to get a grip and wrap around the rod. However, in the case of the rough rod (R_a = 1 mm), the chain easily grips the rod.

5.2 Anisotropic and asymmetric friction

Sawtooth plane. Figure 12 shows a frame from a simulation where a box is interactively being dragged across the surface of a planar surface with an anisotropic normal texture. The texture is generated from a sawtooth displacement profile in which the surface alternates between gentle and steep slopes. The supplementary video shows that as the box is pulled against the steep ridges of the sawtooth pattern, large friction forces are produced and even cause the box to stick until a large enough moment is created by the mouse spring to cause it to topple. Pulling the box against the gentle slope of the ridges also produces friction, but the box begins to eventually slide. Finally, pulling the box along the direction of the ridges results in low friction and the box glides with very little pulling by the mouse spring. The supplementary video additionally shows the effect of changing the saliency parameter k_A that is applied to the plane. As the parameter is gradually
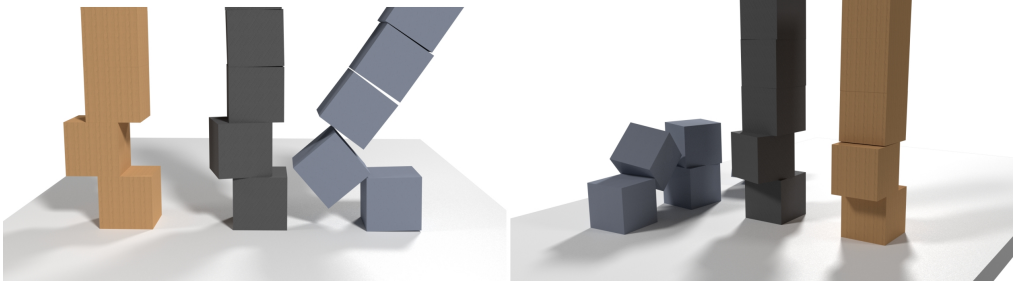
Fig. 14. Box stacks textured with the materials shown in Figure 15 after applying an impulse to the second bottom-most cube in the sideways (left) and forward directions (right). Surface roughness plays a critical role in keeping the stack structure.

increased, forces pulling or pushing against the ridges change from sliding behavior to abrupt sticking behavior.

Crawling worm. The supplementary video demonstrates a functional use of our friction model. A worm is modeled using several box geometries mapped with a sawtooth normal texture. The bodies are connected by distance joints that maintain a specific distance between each body's center of mass. The constraint distance is modulated at run-time using a cosine function to periodically increase and decrease the distance. The combination of asymmetric friction on the worm body and periodic motion of the joints produces an emergent locomotion behavior. The worm slowly inches forward as friction forces propel the head forward during expansion of the joints, whereas the tail is pulled forward during contraction.

Nut and bolt. Figure 13 shows a model of a metric M4 nut and threaded rod, where the threads on the nut and the bolt are modeled in a normal map. With contact frames aligned to the structure direction of the texture (i.e., the threads), we obtain a very low friction for the screw motion of the bolt along the threads and very high friction for all other motions.

Fig. 13. The thread texture on a nut and bolt produces anisotropic friction that only allows a screw motion of the bolt.

5.3 Material textures

Stacking of boxes. We test our friction model using normal textures emulating acrylic, oak, and PVC that has been cut with a tool (see Figure 15). We mapped these normal textures onto the surfaces of cubes arranged in stacks, as shown in Figure 14. All geometrical and physical properties of the cubes are identical, except for the normal textures. Applying a small impulse to the second from bottom cube in each stack demonstrates the effects introduced by our friction model. The acrylic cube is smooth, and the impulse causes it to shoot clear off the stack. The PVC material is the roughest, and moves the least, but is disturbed in such a way that induces some angular motion.
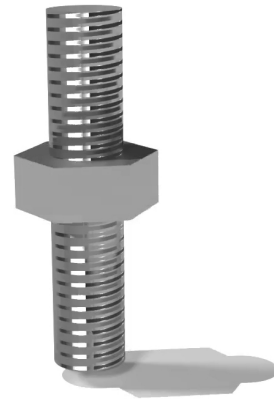
5.4 Friction from thin normal map features

Tumbling bunnies. A thin ridge in the normal map can be seen as local roughness and can produce motion differences with our friction model. Figure 16 shows an example involving three
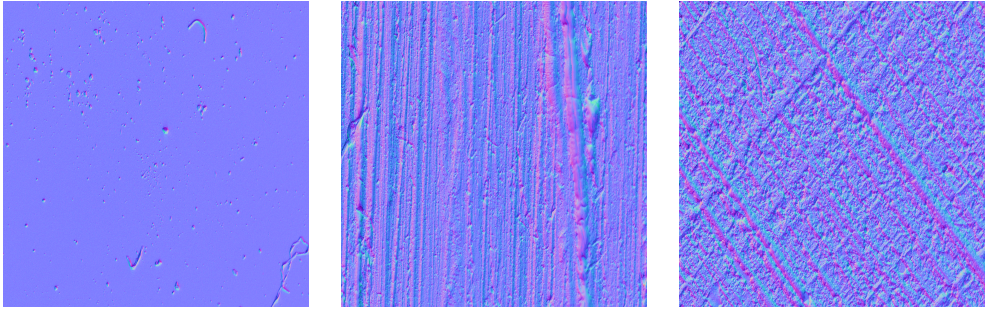
Fig. 15. From left-to-right: normal textures emulating acrylic, oak, and PVC (cut).

Fig. 16. Bunnies tumble down an inclined plane that is embossed with a teapot logo. The rough stone bunny (top) immediately sticks, whereas the metal bunny (middle) and smooth plastic bunny (bottom) slide at different velocities and tumble when hitting the edge of the logo.

bunnies with different roughness that are launched down an inclined surface, which is smooth except for a teapot logo which is embossed on the surface creating ridges. We exaggerate the frictional forces generated by these ridges by increasing the ploughing strength of the surface to $k = 10$. The roughest bunny, which has a stone texture, immediately sticks upon impact causing it to fall over, whereas the metal and smooth plastic bunnies begin to slide down the slope with different velocities, building momentum until they hit the ridge of the embossed teapot, at which point they too begin to tumble.

Table 1. The simulation parameters used in our examples, with corresponding statistics and timings. Computation of Coefficients takes up only a small fraction of the overall dynamics solver time.

	Simulation params.				Performance			
Name	k_A	k_B	μ_0	w	**Average # contacts**	**Dynamics total**	**Contact patch render**	**Compute average coeff.**
Boxes on diff. materials	2.0	2.0	0.2	0.02	16	3.1 ms	0.5 ms	2.3 ms
Sawtooth plane	4.0	1.0	0.2	0.02	4	2.7 ms	0.2 ms	2.3 ms
Chain bolas (rough)	1.0	1.0	0.1	0.02	45	41.6 ms	1.4 ms	6.5 ms
Chain bolas (smooth)	1.0	1.0	0.1	0.02	23	34.1 ms	0.7 ms	3.2 ms
Stacking of boxes	1.0	1.0	0.2	0.02	123	33.1 ms	1.8 ms	2.3 ms
Tumbling bunnies	4.0	1.0	0.1	0.02	10	3.5 ms	0.13 ms	1.7 ms
Nut and bolt	6.0	1.0	0.05	0.004	50	38.1 ms	0.54 ms	2.2 ms
Crawling worm	1.0	1.0	0.1	0.01	94	6.3 ms	0.76 ms	1.9 ms

5.5 Contact frame alignment

For many experiments, the vectors $\vec{t}$ and $\vec{b}$ are aligned with the world coordinate system by default. However, we sometimes found it helpful to align the contact frame with specific local directions. For instance, in the nut and bolt example, one of the tangent directions is aligned with the threading of the bolt by using a callback function. This allows for more precise sampling of the friction limit surface in directions that are critical for achieving the expected behavior. Similarly, in the sawtooth plane example, the friction behavior becomes nearly isotropic if the contact frame is not aligned with, e.g., is diagonal to, the ridges of the texture.

Contact frame alignment is an issue that primarily needs to be addressed when using a box model for friction. Essentially, if the principal axes of the box are not aligned with structural features of the texture, then it fails to produce the expected behavior. One solution to this problem is to perform a preliminary frictionless solve using several iterations of the projected Gauss-Seidel algorithm. This allows the solver to automatically determine good directions for the tangent and binormal basis vectors of the contact frame. Following the frictionless solve, the contact frame is computed such that the first tangent vector $\vec{t}$ aligns with the relative sliding direction at the contact point. Thus, the friction limit surface will be sampled in the direction that objects are likely to move. The supplementary video shows the effect of using automatic contact frame alignment versus a globally aligned frame when the sawtooth plane is rotated 45 degrees around the vertical axis. The example with contact frame alignment demonstrates the expected anisotropy and asymmetry, whereas the motion for the example with fixed globally aligned frames exhibits isotropy.

An alternate solution to the contact frame alignment problem is to increase the number of sample directions. Such a strategy fits well with simulations where the friction limit surface is approximated using a polyhedral cone [Stewart and Trinkle 1996]. However, using our current pipeline, it would require rendering a separate contact patch for every four sample directions and this introduces additional computational overhead.

5.6 Simulation parameters

Table 1 gives the parameters used with our friction model for each of the examples presented in this section. Note that we set $k_A = k_B = k_p$ for all of our examples. The range of values for base coefficient μ_0 is typically quite small. For all simulation results, a frame buffer size of 1024×1024 pixels was used to render and store the 16×16 contact patch sub-textures. Contact patches were rendered using a view plane width according to the fifth column. For comparison, the cubes used in our experiments had a width of 0.20 m. Furthermore, performance and timing metrics are provided. We note that for the more complex examples, rendering the normal textures and computing the average ploughing coefficient for each contact patch typically consumes only

10 − 20% of the total solve time, and the majority of compute time is due to the PGS solver. The PGS solve times we measure are typical to standard friction solves, and we observe that anisotropy and asymmetry do not influence the convergence. Instead, geometric configurations, e.g., stacking, have the largest influence on convergence rates [Erleben 2007]. All of the simulations used a time step of 10 ms, except for the chain bolas and nut and bolt examples, which both used a time step of 1 ms due to the fast movement of the bolas and the precision required for the threading movement.

Evaluation of patch size and resolution. The supplementary video shows a comparison of the tumbling bunnies example and a box on the cast iron plane for different settings of the patch size w and resolution of the contact sub-textures. We observe that when the sub-texture resolution is increased, friction effects due to salient features in the surface may be diminished due to the averaging in Equation 8. Lowering the texture resolution helps to increase the friction effects. However, if the texture is reduced too much, the rendered patches may fail to faithfully capture the roughness of the surface, and thus the frictional behavior is more slippery. Regarding the patch size, increasing w can similarly diminish frictional effects since the spatial resolution of the sub-texture reduces as the patch size increases. This also manifests as more slippery frictional behavior. However, for small patches, we also observed that especially for sliding motions, salient features may be missed due to the discrete collision detection, which can be seen in the tumbling bunnies comparison.

6 CONCLUSIONS

This work proposes a friction modeling and simulation pipeline that is capable of simulating a wide variety of friction phenomena such as isotropy, anisotropy, asymmetry, and non-convexity. Our GPU based sampling technique can be seen as an on-the-fly computation that generates a general limit surface [Goyal et al. 1989], i.e., a friction cone. As such, our approach is compatible with many existing simulation pipelines.

A further benefit of our approach is that we do not require crafting and storing a table of friction coefficients. Only the meso-facet geometry is required to compute friction, although we do allow using a base coefficient that accounts for nominal friction effects. Furthermore, compared to previous work, our approach does not require that surfaces having uniform structural directions, and even out-of-plane directions are considered when computing friction. We demonstrate the behavior produced by our model using normal maps generated from statistical distributions, as well as ones scanned from real-world example surfaces using photogrammetric techniques.

Our demonstrations are limited to normal mapping techniques since they are well-suited to an interactive simulation setting, but they also draw a strong connection to the micro-geometry of rough surfaces. However, displacement mapping is an alternative technique that could also be employed for friction simulation. A possible advantage that may come from using displacement maps is that it could provide more accurate estimates of where asperities come into contact. For instance, by integrating over the peaks of a displacement map in the colliding regions between two surfaces, this gives an estimate of the real contact area, which may then combined with the apparent area from the macro geometry in order to compute a friction coefficient based on standard tribology models [Bowden and Tabor 1950; Sheng Chen and Liu 2016]. Yet in the context of interactive computer graphics applications, displacement mapping is less prolific due to the overhead associated with the modifications to surface geometry. As our work shows, this level of detail is not necessary for rich, real-time friction effects.

While the friction model presented in this paper is based on a tribology ploughing model, our approach diverges from the fundamental model in several aspects. We extend the basic ploughing model with additional terms that modulate friction based on the relative sliding direction and facet

orientation. We also propose an additive friction model based on an assumption that the real area of contact is small, and thus compute a friction coefficient separately for each surface based on asperity interlocking. These changes produce behavior that is qualitatively plausible and leads to an efficient simulation pipeline.

6.1 Limitations and future work

Currently, our work is limited to very stiff and rough surfaces, and hence is suitable for rigid body simulations. Furthermore, we assume perfectly elastic interactions with no micro-fracturing and plastic effects (e.g., wear and tear). This results in deterministic behavior when repeating an action in a game or simulator, as is often desirable.

If soft materials were to be considered, then both plastic and elastic deformations of asperities could be important. However, we do believe our approach could be extended to a soft body simulator or different types of simulator paradigms as long as the contact solver is based on a friction cone (e.g., see [Erleben et al. 2020]).

The current numerical approach we presented has been tailored for speed and performance in a real-time physics engine, since we target games and other interactive applications. We exploit the rendering pipeline to quickly sample and evaluate our friction model on a per contact patch basis. In less time-critical systems, one may wish to evaluate the friction model over the whole apparent contact area or deploy different sampling strategies than ours. This is outside the scope of our target applications, and may also present some challenges for seamless integration into a rendering pipeline.

Although we use GPU rendering to improve the efficiency of sampling the surface around each contact, it does present some drawbacks. Mainly that the total number of contacts in the simulation is limited by the number of sub-textures that can be stored by the frame buffer we use to render the contact patches. A workaround is to simply increase the size of the frame buffer. However, transferring a larger frame buffer to the CPU memory, or doing more frequent transfers, would diminish performance. This is an aspect of our pipeline we intend to optimize.

Finally, the ability of our model to accurately predict real-world behavior remains to be evaluated. However, there are interesting and important applications, particularly in the context of robotics, for realistic friction models that derive their behavior directly from surface roughness. For instance, predicting frictional properties based on signals from vision-based sensor would be possible with our model if roughness could be estimated from surface appearance. We believe this opens new possibilities for robotic learning, especially in environments where physical properties of objects are not known in advance.

ACKNOWLEDGMENTS

REFERENCES

Jérémie Allard, François Faure, Hadrien Courtecuisse, Florent Falipou, Christian Duriez, and Paul G. Kry. 2010. Volume Contact Constraints at Arbitrary Resolution. *ACM Trans. Graph.* 29, 4, Article 82 (July 2010), 10 pages. https://doi.org/10.1145/1833349.1778819

Sheldon Andrews and Kenny Erleben. 2021. Contact and Friction Simulation for Computer Graphics. In *ACM SIGGRAPH 2021 Courses* (Virtual Event, USA) *(SIGGRAPH '21).* Association for Computing Machinery, New York, NY, USA, Article 2, 124 pages. https://doi.org/10.1145/3450508.3464571

David Baraff. 1994. Fast Contact Force Computation for Nonpenetrating Rigid Bodies. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*. ACM, New York, NY, USA, 23–34. https://doi.org/10.1145/192161.192168

Jan Bender, Kenny Erleben, and Jeff Trinkle. 2014. Interactive Simulation of Rigid Body Dynamics in Computer Graphics. *Computer Graphics Forum* 33, 1 (2014), 246–270. https://doi.org/10.1111/cgf.12272

Frank Philip Bowden and David Tabor. 1950. *The friction and lubrication of solids*. Oxford : Clarendon Press ; New York : Oxford University Press. https://doi.org/10.1119/1.1933017

Zhili Chen, Renguo Feng, and Huamin Wang. 2013. Modeling Friction and Air Effects Between Cloth and Deformable Bodies. *ACM Trans. Graph.* 32, 4, Article 88 (July 2013), 8 pages. https://doi.org/10.1145/2461912.2461941

Jonathan Cohen, Marc Olano, and Dinesh Manocha. 1998. Appearance-Preserving Simplification. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 115–122. https://doi.org/10.1145/280814.280832

Antoine Costes, Fabien Danieau, Ferran Argelaguet, Anatole Lécuyer, and Philippe Guillotel. 2018. Haptic Material: A Holistic Approach for Haptic Texture Mapping. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer, 37–45.

Gilles Daviet. 2020. Simple and Scalable Frictional Contacts for Thin Nodal Objects. *ACM Trans. Graph.* 39, 4, Article 61 (July 2020), 16 pages. https://doi.org/10.1145/3386569.3392439

Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. *ACM Trans. Graph.* 30, 6, Article 139 (Dec. 2011), 12 pages. https://doi.org/10.1145/2070781.2024173

Keno Dreßel, Kenny Erleben, Paul G. Kry, and Sheldon Andrews. 2019. Automated Acquisition of Anisotropic Friction. In *16th Conference on Computer and Robot Vision (CRV 2019)*. 159–165. https://doi.org/10.1109/CRV.2019.00029

Kenny Erleben. 2007. Velocity-based Shock Propagation for Multibody Dynamics Animation. *ACM Trans. Graph.* 26, 2, Article 12 (June 2007), 20 pages. https://doi.org/10.1145/1243980.1243986

Kenny Erleben. 2017. Rigid Body Contact Problems Using Proximal Operators. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '17)*. Association for Computing Machinery, New York, NY, USA, Article 13, 12 pages. https://doi.org/10.1145/3099564.3099575

Kenny Erleben, Miles Macklin, Sheldon Andrews, and Paul G. Kry. 2020. The Matchstick Model for Anisotropic Friction Cones. *Computer Graphics Forum* 39, 1 (Feb. 2020), 450–461. https://doi.org/10.1111/cgf.13885

Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. 2020. ADD: Analytically Differentiable Dynamics for Multi-Body Systems with Frictional Contact. *ACM Trans. Graph.* 39, 6, Article 190 (Nov. 2020), 15 pages. https://doi.org/10.1145/3414685.3417766

Suresh Goyal, Andy Ruina, and Jim Papadopoulos. 1989. Limit Surface and Moment Function Descriptions of Planar Sliding. In *Proc. of the 1989 IEEE International Conference on Robotics and Automation (Vol. 2)*. Scottsdale, AZ, 794–799. https://doi.org/10.1109/ROBOT.1989.100081

James A. Greenwood and J. B. P. Williamson. 1966. Contact of Nominally Flat Surfaces. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 295, 1442 (1966), 300–319. https://doi.org/10.1098/rspa.1966.0242

Danny M. Kaufman, Timothy Edmunds, and Dinesh K. Pai. 2005. Fast Frictional Dynamics for Rigid Bodies. *ACM Trans. Graph.* 24, 3 (July 2005), 946–956. https://doi.org/10.1145/1073204.1073295

Danny M. Kaufman, Shinjiro Sueda, Doug L. James, and Dinesh K. Pai. 2008. Staggered Projections for Frictional Contact in Multibody Systems. *ACM Trans. Graph.* 27, 5, Article 164 (Dec. 2008), 11 pages. https://doi.org/10.1145/1409060.1409117

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection-and Inversion-Free, Large-Deformation Dynamics. *ACM Trans. Graph.* 39, 4, Article 49 (July 2020), 20 pages. https://doi.org/10.1145/3386569.3392425

Martha Liley, Delphine Gourdon, Dimitrios Stamou, Ulrich Meseth, Thomas M. Fischer, Carsten Lautz, Henning Stahlberg, Horst Vogel, Nancy A. Burnham, and Claus Duschl. 1998. Friction anisotropy and asymmetry of a compliant monolayer induced by a small molecular tilt. *Science* 280, 5361 (1998), 273–275. https://doi.org/10.1126/science.280.5361.273

Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. 2020. Projective Dynamics with Dry Frictional Contact. *ACM Trans. Graph.* 39, 4, Article 57 (July 2020), 8 pages. https://doi.org/10.1145/3386569.3392396

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019. Non-Smooth Newton Methods for Deformable Multi-Body Dynamics. *ACM Trans. Graph.* 38, 5 (2019), 20.

Miguel A. Otaduy, Nitin Jain, Avneesh Sud, and Ming C. Lin. 2004. Haptic display of interaction between textured models. In *IEEE Visualization 2004*. IEEE, 297–304. https://doi.org/10.1109/VISUAL.2004.37

Miguel A. Otaduy, Rasmus Tamstorf, Denis Steinemann, and Markus Gross. 2009. Implicit Contact Handling for Deformable Objects. *Computer Graphics Forum* 28, 2 (2009), 559–568. https://doi.org/10.1111/j.1467-8659.2009.01396.x

Simon Pabst, Bernhard Thomaszewski, and Wolfgang Straßer. 2009. Anisotropic Friction for Deformable Surfaces and Solids. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New Orleans, Louisiana)

(SCA '09). ACM, New York, NY, USA, 149–154. https://doi.org/10.1145/1599470.1599490

Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond, and Som H. Yau. 2001. Scanning Physical Interaction Behavior of 3D Objects. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. ACM, New York, NY, USA, 87–96. https://doi.org/10.1145/383259.383268

Albert Peiret, Sheldon Andrews, József Kövecses, Paul G. Kry, and Marek Teichmann. 2019. Schur Complement-Based Substructuring of Stiff Multibody Systems with Contact. *ACM Trans. Graph.* 38, 5, Article 150 (Oct. 2019), 17 pages. https://doi.org/10.1145/3355621

Ernest Rabinowicz. 1995. *Friction and Wear of Materials*. Wiley.

Zhimin Ren, Hengchin Yeh, and Ming C Lin. 2010. Synthesizing contact sounds between textured models. In *2010 IEEE Virtual Reality Conference (VR)*. IEEE, 139–146. https://doi.org/10.1109/VR.2010.5444799

Shmuel M. Rubinstein, Gil Cohen, and Jay Fineberg. 2006. Contact Area Measurements Reveal Loading-History Dependence of Static Friction. *Phys. Rev. Lett.* 96 (Jun 2006), 256103. Issue 25. https://doi.org/10.1103/PhysRevLett.96.256103

Gang Sheng Chen and Xiandong Liu. 2016. Chapter 3 - Friction. In *Friction Dynamics*, Gang Sheng Chen and Xiandong Liu (Eds.). Woodhead Publishing, 91 – 159. https://doi.org/10.1016/B978-0-08-100285-8.00003-1

David E Stewart and Jeffrey C Trinkle. 1996. An Implicit Time-stepping Scheme for Rigid Body Dynamics with Inelastic Collisions and Coulomb Friction. *Internat. J. Numer. Methods Engrg.* 39, 15 (1996), 2673–2691. https://doi.org/10.1002/(SICI)1097-0207(19960815)39:15<2673::AID-NME972>3.0.CO;2-I

Paul Umbanhowar, Thomas H. Vose, Atsushi Mitani, Shinichi Hirai, and Kevin M. Lynch. 2012. The effect of anisotropic friction on vibratory velocity fields. In *2012 IEEE International Conference on Robotics and Automation*. 2584–2591. https://doi.org/10.1109/ICRA.2012.6225273

Antonis I. Vakis, Vladislav A. Yastrebov, Julien Scheibert, Lucia Nicola, Daniele Dini, Clotilde Minfray, Andreas Almqvist, Marco Paggi, Seunghwan Lee, Georges Limbert, Jean-François Molinari, Guillaume Anciaux, Ramin Aghababaei, Sebastián Echeverri Restrepo, Antonio Papangelo, Antonio Cammarata, Paolo Nicolini, Carmine Putignano, Giuseppe Carbone, Stanisław Stupkiewicz, Jakub Lengiewicz, Gianluca Costagliola, Federico Bosia, Roberto Guarino, Nicola M. Pugno, Martin H. Müser, and Michele Ciavarella. 2018. Modeling and simulation in tribology across scales: An overview. *Tribology International* 125 (2018), 169 – 199. https://doi.org/10.1016/j.triboint.2018.02.005

Kees van den Doel, Paul G. Kry, and Dinesh K. Pai. 2001. FoleyAutomatic: Physically-Based Sound Effects for Interactive Simulation and Animation. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 537–544. https://doi.org/10.1145/383259.383322

Simon V. Walker and Remco I. Leine. 2017. Anisotropic dry friction with non-convex force reservoirs: modeling and experiments. In *9th European Nonlinear Dynamics Conference (ENOC 2017)*. 1–7.

Chengjiao Yu and Q Jane Wang. 2012. Friction anisotropy with respect to topographic orientation. *Scientific reports* 2, 1 (2012), 1–6. https://doi.org/10.1038/srep00988